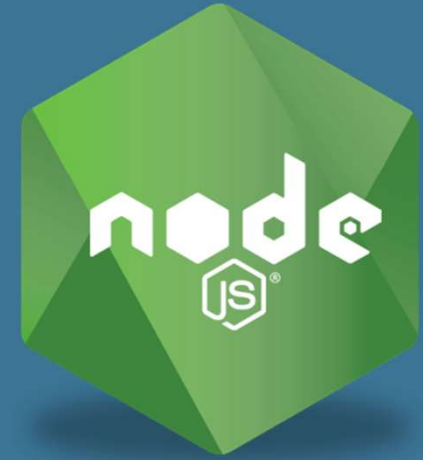# తెలుగులో

PART – 2

➢ **Modules**

➢ **Export, Require**

➢ **Import, Export**

Venkatesh Mogili
#WebGuru

Crash Course

Contents

- **What are modules in NodeJS?**
- How to export and import modules?
- ES6 syntax to export and import modules.

Venkatesh Mogili
   #WebGuru

Modules

➢ **What are modules in NodeJS?**

✓ **Module** in Node.js is **a simple or complex functionality organized in single or multiple JavaScript files** which can be reused throughout the Node.js application.

✓ Each **module** in Node.js has its own context, so it cannot interfere with other **modules** or pollute global scope.

✓ **module** keyword used to represent the current module.

Contents

- ✓ What are modules in NodeJS?
- ➢ **How to export and import modules?**
- ➢ ES6 syntax to export and import modules.

Venkatesh Mogili
#WebGuru

Export and Require

➤ How to export and import modules?

Export: (Literals)

✓ exports is an object, we can assign anything to it. It will consider the exported one as module whether it's a string or function or variable anything.

✓ **module.exports = module;**

✓ **Ex: module.exports ='Hello World!';**

Import:

✓ **var output = require('./filename.js');**
console.log(output); // Hello World!

filename is a file which we are exporting our module with relative path using **./**

✓ **require('http') , require('mysql'), require('express')**
http, mysql, express are npm modules, that's why we imported without **./**

Venkatesh Mogili
#WebGuru

Export and Require

➢ **How to export and import modules?**

Export: (Object-Properties)

✓ exports is an object, we can assign properties and methods to it.

✓ **module.exports.objectName = module; OR exports.objectName=module;**

✓ **Ex: module.exports.sample ='Hello World!'; OR exports.sample='Hello World!';**
sample is a property here.

Import:

✓ **var output = require('./filename.js');**
console.log(output.sample); //Hello World!

filename is a file which we are exporting our module with relative path using **./**

Export and Require

➤ **How to export and import modules?**

Export: (Object-Methods)

✓ exports is an object, we can assign anything to it. It will consider the exported one as module whether it's a string or function or variable anything.

✓ **module.exports.objectName = module;**

✓ **Ex:**
module.exports.log = function (msg){
    console.log(msg);
};

  log is a method here.

Import:

✓ **var output = require('./filename.js');**
output.log('Hello World!') // Hello World!

  filename is a file which we are exporting our module with relative path using **./**

Venkatesh Mogili
#WebGuru

Export and Require

➢ **How to export and import modules?**

Import:

✓ **var output = require('./foldername');**
foldername contains several modules or single module exported.

Ex:
var output = require('./utility');
utility is a folder which is exporting a log module.

✓ It will search for **package.json** file in that specified folder and **"main"** entry.

✓ If package.json file doesn't exist, then it will look for **index.js** file as a module.

Venkatesh Mogili
#WebGuru

Contents

- ✓ What are modules in NodeJS?
- ✓ How to export and import modules?
- ➢ **ES6 syntax to export and import modules.**

Venkatesh Mogili
   #WebGuru

ES6 - Export and Import

➢ **ES6 syntax to export and import modules.**

Export:

✓ **export default moduleName;**   // Default Export

✓ **export {moduleName};**   // Named Export

Import:

✓ **import moduleName from './module-path';**
   OR
   import moduleName as otherName from './module-path'; // Default Import

✓ **import {moduleName} from './module-path';**   // Named Import

ES6 - Export and Import

- ➢ **ES6 syntax to export and import modules?**
- ➢ You cannot directly use ES6 export and import features. It will throw an error as **SyntaxError: Cannot use import statement outside a module**
- ✓ Need to mention **"type"** property in package.json as **"module".**

- ➢ If you use commonjs syntax along with es6 import syntax then it will throw an error as **ReferenceError: require is not defined**
- ✓ import module from "module";
- ✓ const require = module.createRequire(import.meta.url);
- ✓ const output = require("./log.cjs");

// cjs extension supports commonjs. mjs/js extension supports es6.

OR

// js extension supports both If we create module folder with package.json and main entry.

- ✓ Now, we can use both commonjs(require) and es6(import).

Contents

- ✓ What are modules in NodeJS?
- ✓ How to export and import modules?
- ✓ ES6 syntax to export and import modules.

Venkatesh Mogili
#WebGuru

Summary

✓ **Module** is a simple or complex functionality organized in single or multiple JavaScript files which can be reused throughout the Node.js application.

✓ **module.exports** used for export and **require** used for importing in ES Modules.

✓ **export** and **import** keywords used for ES6 Modules.
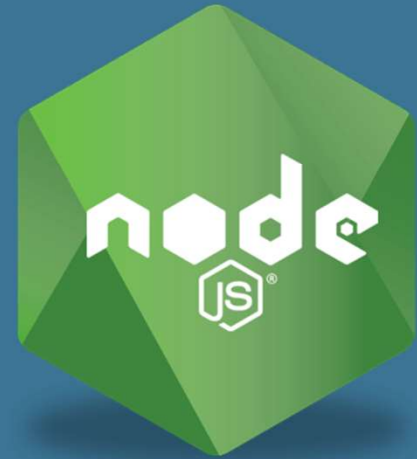
Venkatesh Mogili
#WebGuru

Sharing Is Caring

SUBSCRIBE

LIKE SHARE COMMENT